

# DIFFERENCE IN MONOTONICITY BETWEEN P AND NP

KOBAYASHI, KOJI

## 1. OVERVIEW

This paper talks about monotonicity of P and NP by using monotone circuit and Log Space Transducer (LST).

We can reduce Deterministic Turing Machine (DTM) to monotone circuit value problem by using LST. Therefore DTM has nonmonotonicity of the degree that LST can compute it. But Nondeterministic Turing Machine (NTM) have nonmonotonicity that LST cannot compute it. Therefore We cannot reduce NP-Complete problem to P-Complete problem by using LST.

The NP-Complete problem is different from P-Complete problem in non-monotonicity. And NP-Complete is not in P-Complete.

## 2. NONMONOTONICITY OF LST

First, I show nonmonotonicity of LST. LST have read only input tape, write only output tape, read and write working tape that is logarithm size, head, and state register that is constant size. The information of working tape, head position, and state registers is replaceable and become non-monotonicity as information.

**Definition 1.** I will use the term “Nonmonotonicity information” as the total information of working tape, head position, and state registers. I will use the term “Monotonic information” as the total information of input tape and output tape.

**Theorem 2.** *Nonmonotonic information size of LST is at most logarithm size.*

*Proof.* We can record working tape, head position and state register in logarithm size, logarithm size pointer, constant size of record. Therefore, we can record all of nonmonotonic information in logarithm size.  $\square$

P-Complete problem have some nonmonotonic information. But We can reduce P-Complete problem to monotone information by using LST.

**Definition 3.** I will use the term “Flatten input” as changing each value of an input (Target input) of problem (Target problem) to

$$\begin{cases} 0 \rightarrow 10 \\ 1 \rightarrow 01 \end{cases}$$

And I will use the term “Flatten problem” as that problem that reduce object input to flatten input, “Flatten transducer” as transducer that change target input to flatten input.

LST can change target input to flatten input.

**Theorem 4.** *We can make Flatten transducer with LST.*

*Proof.* Flatten transducer have read only input, write only output, pointer of input position, and state register that identify where flatten input value outputting. We can make this transducer with LST.  $\square$

### 3. MONOTONICITY OF DTM

I show Monotonicity of DTM by showing P-Completeness of value decide problem of monotone circuit.

**Definition 5.** I will use the term “*MONOTONE – VALUE*” as the *CIRCUIT – VALUE* of monotone circuit.

**Theorem 6.** *MONOTONE – VALUE is P-Complete.*

*Proof.* *CIRCUIT – VALUE* include *MONOTONE – VALUE* therefore *MONOTONE – VALUE* is in P.

Next, I show reduction all P problem to *MONOTONE – VALUE*. I show this reduction by using DTM. This reduction is two steps. First step is that transducer change target input to flatten input. As a mentioned above 4, We can change input with LST. Second step is that reduce DTM computing flatten input to *MONOTONE – VALUE*. As a mentioned in Theorem 9.30 of [1], *CIRCUIT – VALUE* that emulate DTM have only Not gate in input flattening. Therefore, we can reduce *CIRCUIT – VALUE* that compute flatten problem to *MONOTONE – VALUE* that have no Not gate. And we can reduce to *MONOTONE – VALUE* with LST same as *CIRCUIT – VALUE* reduction. Therefore we can reduce P-Complete problem to *MONOTONE – VALUE* with these two step LST.

Therefore, *MONOTONE – VALUE* is P-Complete.  $\square$

That is, P-Complete problem have only nonmonotonicity that can compute with LST.

**Theorem 7.** *We can reduce P-Complete problem to monotone problem with LST.*

*Proof.* Mentioned above 6, We can reduce all P-Complete problem to *MONOTONE – VALUE* with LST. *MONOTONE – VALUE* is monotone problem. Therefore we can reduce all P-Complete problem to monotone problem with LST.  $\square$

### 4. NONMONOTONICITY OF NTM

I show Nonmonotonicity of NTM.

**Theorem 8.** *We cannot reduce NTM to monotone problem with LST.*

*Proof.* I prove it using reduction to absurdity. We assume that we can reduce NTM to monotone problem with LST. Mentioned above 2, LST can record at most logarithm size of nonmonotonic information. Therefore, NTM have at most logarithm size of nonmonotonic information.

But, NTM can record polynomial number of configuration. And NTM cancel these configuration if these configuration yield to rejecting configuration, therefore we cannot treat these information as monotonic information. That is, we must treat these configuration nonmonotonic information. We must compute these non-monotonic information to reduce NTM to monotonic problem. But We can treat at most logarithm size of nonmonotonic information with LST and we cannot treat

polynomial number of configuration with LST and contradicts a condition that treat LST.

Therefore, this theorem was shown than reduction to absurdity.  $\square$

We cannot reduce NTM to monotone problem. Therefore, we cannot reduce NTM to *MONOTONE – VALUE*. And NTM is not P.

**Theorem 9.** *NTM is not P.*

*Proof.* Mentioned above 8, we cannot reduce NTM to monotone problem with LST. Therefore, we cannot reduce NTM to *MONOTONE – VALUE* with LST. And mentioned above 6, *MONOTONE – VALUE* is P-Complete problem. Therefore, NTM is not P.  $\square$

#### REFERENCES

- [1] Michael Sipser, (translation) OHTA Kazuo, TANAKA Keisuke, ABE Masayuki, UEDA Hiroki, FUJIOKA Atsushi, WATANABE Osamu, Introduction to the Theory of COMPUTATION Second Edition, 2008
- [2] HAGIYA Masami, NISHIZAKI Shinya, Mechanism of Logic and Calculation, 2007